

第3章 CSS 基础

CSS 是一种实现网页内容格式化的一种技术。在前面 HTML 相关知识的学习中，我们知道可以通过标记实现对网页内容进行简单样式处理，如文字大小、排列方式等。但是要对网页做更多丰富的风格处理则需要用到 CSS。

3.1 CSS 简介

CSS (Cascading Style Sheets) 即级联式风格纸，简称样式表。样式是将网页内容和内容风格分离的一种语言，能够对 HTML 控制的内容进行格式的控制，如文字的大小、段落的宽度、列表的标记等。级联，也称层叠，是指当 HTML 网页文件引用了多个 CSS 样式时，如果 CSS 的规则定义产生了冲突，浏览器将根据层次的先后顺序来应用样式，一般会遵守“就近优先”原则。

根据前面章节的介绍，我们知道使用 HTML 制作网页并不难，但在进行网站制作时会涉及很多页面，如果想对页面的一些内容进行统一风格的制定，则不得不对每张网页的标记进行更改，显然工作量是很大的。使用 CSS 就可以解决这一问题，因为 CSS 可将样式的定义与 HTML 网页内容分离，只要建立一个 CSS 文件定义样式，再让所有的 HTML 网页文件调用此 CSS 文件的样式即可。如果以后需要改变 HTML 文件中的内容样式，只需对 CSS 文件样式规则进行调整，就可以改变网页内容样式。

CSS 功能很强大，能够控制几乎所有的网页元素，进行丰富的风格定义。它不仅可以在 HTML 中使用，还可以用于 XML 和 JavaScript 中。现在几乎所有漂亮的网页都会用 CSS，它已经成为网页设计中必不可少的工具之一。

3.2 CSS 基本语法

3.2.1 CSS 的基本结构

CSS 是一个包含若干个规则的文件，这些规则通过属性和值控制网页中某个元素的显示方式。属性可以控制各种元素，如文本大小、颜色、元素位置以及元素的显示或关闭等。

1. CSS 规则结构

样式表的每一个规则由两个部分组成：选择器 (selector) 和声明 (declaration)。选择器确定网页中哪个内容受控制，声明则由一个或多个属性值对组成。其基本语法如下：

```
selector{属性 1:属性值 1; 属性 2:属性值 2;……属性 n:属性值 n}
```

说明：

(1) selector 表示希望进行格式化的元素，可以是 HTML 标签，也可以是类或其他类型的选择器。

(2) 声明部分必须放置在选择器后的大括号内。

(3) 属性和值之间以冒号进行间隔，属性值对之间以分号进行间隔。

2. 注释

在样式表中需要添加注释来帮助设计者记住复杂的样式规则的应用范围和作用，便于维护和应用。以“//”实现单行注释，“/*注释内容*/”实行多行注释。

3. 示例

对块级元素段落中的文字进行风格设置，文字大小为 12 像素，文字颜色为红色，其格式如下：

```
P{
  font-size:12px;
  color:red
} //对段落文字进行大小和颜色设置
```

3.2.2 CSS 规则定义

CSS 的规则定义，即对网页内容和结构进行样式属性设置，如文本属性、背景属性、边框属性和定位属性等。

1. 文本属性

文本属性包括：

- **font-family:** 设置样式字体。
- **font-size:** 设置字体大小。可以通过选择数字和度量单位来选择特定的大小，也可以选择相对大小。
- **font-weight:** 设置字体粗细效果。“正常”等于 400，“粗体”等于 700。
- **font-style:** 指定“正常”“斜体”或“偏斜体”作为字体样式，默认设置为“正常”。
- **font-variant:** 设置文本的小型大写字母变体。
- **line-height:** 设置文本所在行的高度。习惯上将该设置称为行高。选择“正常”自动计算字体大小的行高，或输入一个确切的值并选择一种度量单位。两种浏览器都支持行高属性。
- **text-transform:** 将所选内容中每个单词的首字母大写或将文本设置为全部大写或小写。
- **text-decoration:** 向文本中添加下划线、上划线或删除线，或使文本闪烁。常规文本的默认设置是“无”。链接的默认设置是“下划线”。将链接设置设为无时，可以通过定义一个特殊的类去除链接中的下划线。
- **color:** 设置文本颜色。

2. 背景属性

背景属性用于定义 CSS 样式的背景设置。可以对网页中的任何元素应用背景属性。例如，创建一个样式，将背景颜色或背景图像添加到任何页面元素中，比如在文本、表格、页面等的后面。包括：

- **background-image:** 设置元素的背景图片。
- **background-repeat:** 确定是否以及如何重复背景图像。“不重复”是只在元素开始处显示一次图像。“重复”是在元素的后面水平和垂直平铺图像。“横向重复”和“纵向重复”是分别显示图像的水平带区和垂直带区。图像被剪辑以适合元素的边界。

- **background-attachment**: 确定背景图像是固定在其原始位置还是随内容一起滚动。注意, 某些浏览器可能将“固定”选项视为“滚动”。Internet Explorer 支持该选项, 但 Netscape Navigator 不支持。
- **background-position(X)**和 **background-position(Y)**: 指定背景图像相对于元素的初始位置。这可用于将背景图像与页面中心垂直(Y)和水平(X)对齐。如果附件属性为“固定”, 则位置相对于“文档”窗口而不是元素。

3. 区块属性

区块属性用于定义标签和属性的间距和对齐设置。包括:

- **word-spacing**: 设置字词的间距。若要设置特定的值, 请在弹出菜单中选择“值”选项, 然后输入一个数值。在第二个弹出菜单中选择度量单位(如像素、点等)。
- **letter-spacing**: 增加或减小字母或字符的间距。若要减小字符间距, 请指定一个负值(如-4)。字母间距设置覆盖对齐的文本设置。
- **vertical-align**: 指定应用此属性的元素的垂直对齐方式。Dreamweaver 仅在将此属性应用于 标签时, 才会在“文档”窗口中显示。
- **text-align**: 设置文本在元素内的对齐方式。
- **text-indent**: 指定第一行文本缩进的程度。可以使用负值创建凸出, 但显示方式取决于浏览器。
- **white-space**: 确定如何处理元素中的空格。从三个选项中进行选择: “正常”, 收缩空白; “保留”, 其处理方式与文本被放在 pre 标签中一样(即保留所有空白, 包括空格、制表符和回车); “不换行”, 指定仅当遇到 br 标签时文本才换行。
- **display**: 指定是否及如何显示元素。“无”指定到某个元素时, 将禁用该元素的显示。

4. 方框属性

方框属性可以用于控制元素在页面上的放置方式的标签和属性定义设置。可以在应用填充和边距设置时将设置应用于元素的各个边, 也可以使用“全部相同”设置将相同的设置应用于元素的所有边。包括:

- **width** 和 **height**: 设置元素的宽度和高度。
- **float**: 指定其他元素在浮动的元素周围流动的一侧。浮动的元素固定在浮动一侧, 其他内容在另一侧围绕它流动。
- **clear**: 指定不允许有其他浮动元素的元素一侧。
- **padding**: 填充。指定元素内容与元素边框之间的间距(如果没有边框, 则为边距)。取消选择“全部相同”选项可设置元素各个边的填充。
- **margin**: 边距。指定一个元素的边框与另一个元素之间的间距(如果没有边框, 则为填充)。

5. 边框属性

边框属性用于定义元素周围边框的设置(宽度、颜色和样式)。包括:

- **style**: 设置边框的样式外观。样式的显示方式取决于浏览器。取消选择“全部相同”可设置元素各个边的边框样式。
- **width**: 设置元素边框的粗细。
- **color**: 设置边框的颜色。可以分别设置每条边的颜色, 取消选择“全部相同”复选项可设置元素各个边的边框颜色。

6. 列表属性

列表属性用于定义列表设置，如项目符号大小和类型。包括：

- **list-style-type**: 设置列表符号或编号的外观。
- **list-style-image**: 为项目符号指定自定义图像。单击“浏览”选择图像，或键入图像的路径。
- **list-style-position**: 设置列表项文本是否换行并缩进（外部），或者文本是否换行到左边距（内部）。

7. 定位属性

定位属性用于确定与选定的 CSS 样式相关的内容在页面上的定位方式。包括：

- **position**: 位置。确定浏览器应如何定位选定的元素。
- **visibility**: 可见性。确定内容的初始显示条件。如果不指定可见性属性，则默认情况下，内容将继承父级标签的值。
- **z-index**: Z 轴。确定内容的堆叠顺序。Z 轴值较高的元素显示在 Z 轴值较低的元素（或根本没有 Z 轴值的元素）上方。值可以为正，也可以为负。
- **overflow**: 溢出。确定当容器（如 Div 或 P）的内容超出容器的显示范围时的处理方式。
- **placement**: 位置。指定内容块的位置和大小。浏览器如何解释位置取决于“类型”设置。如果内容块的内容超出指定的大小，则将改写大小值。
- **clip**: 剪辑。定义内容的可见部分。如果指定了剪辑区域，则可以通过脚本语言（如 JavaScript）访问它，并操作属性以创建像擦除这样的特殊效果。

8. 其他属性

其他属性还有：滤镜、分页和指针选项。

- **page-break-before**: 打印期间在样式所控制的对象之前强行分页。
- **page-break-after**: 打印期间在样式所控制的对象之后强行分页。
- **cursor**: 当指针位于样式所控制的对象上时改变指针图像。
- **filter**: 对样式所控制的对象应用特殊效果。

3.3 CSS 选择器分类

要使用 CSS 对 HTML 页面中的元素实现一对一、一对多甚至多对一的控制，就需使用 CSS 选择器。HTML 页面中的元素都可通过 CSS 选择器进行控制。

设计者可以选择不同类型的选择器，来灵活控制页面 CSS 样式的应用。

3.3.1 标签选择器

一个完整的 HTML 页面由很多不同的标签（标记）组成，而标签选择器则是决定哪些标签采用哪种 CSS 样式。比如要实现段落中文字大小为 12 像素，文字颜色为蓝色，文字字体为黑体，其 CSS 编写规则如下：

```
P{font-size:12px;color:blue;font-family:黑体}
```

练一练: 编写标签选择器设置 CSS 风格，实现由图 3-1 转化为图 3-2 所示的界面效果。页面解读：“学院新闻”为标题标签<h2>，具体新闻列表是无序列表标签 ul。



图 3-1 学院新闻未风格化界面



图 3-2 学院新闻风格化后界面

其关键代码如下：

```
a {
    text-decoration: none;
    color: #666666;
    float: left;
}
ul {
    list-style-type: none;
    font-size: 12px;
    margin-left: 20px;
    padding: 0px;
}
```

```

li {
    line-height: 30px;
    text-align: right;
    color: #666666;
}

```

3.3.2 类名选择器

在 HTML 中，标签元素可以定义属性类 `class`，类名可应用于 HTML 文档中不同标签的元素中，如果要实现不同标签标记的内容具有相同的 CSS 风格，可使用类名选择器，当类名作为选择器时，需使用小数点进行标识。

例如：

```

<h2 class="style1">学院新闻</p>
<ul>
<li class="style1">秦皇岛校外学习中心</li>
</ul>

```

如果要实现类名为“style1”的标签所控制的内容字体类型为宋体，CSS 编写规则如下：

```

.style{
    Font-family:宋体
}

```

3.3.3 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。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“school”，那么在同一页面就不能再将其他元素 ID 取名为“school”了。即便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，对于后继学习的 JS 而言，它只会选择具有相同 ID 名字元素中的第一个。培养一个好的编程习惯，同一个 ID 不要在页面中出现两次。

ID 作为选择器时，需使用“#”进行标识。如对学院新闻标题和新闻列表设置不同的字体大小，其关键代码如下：

```

<style type="text/css">
#newsid{
    font-size:36px
}
#newslist{
    font-size:12px
}
</style>
.....
<h2 id="newsid">学院新闻</p>
<ul>
<li id="newslist">秦皇岛校外学习中心</li>
</ul>

```

3.3.4 伪类选择器

当用户浏览网页时，会发现访问过的超链接、未访问的超链接和正准备点击的超链接，会有不同的外观显示，如图 3-3 所示，当鼠标指向超链接时，文字颜色发生了改变。



图 3-3 鼠标指向超链接状态

伪类并非是真的类，它们是在浏览器中具有一定意义的预定义类。使用伪类选择器可设置超链接四种状态的 CSS 风格特征，其作用如下：

- (1) A:link，面向未访问的链接。
- (2) A:hover，面向指向链接。
- (3) A:active，面向激活的链接（即正在点击的链接）。
- (4) A:visited，面向访问过的链接。

实现图 3-3 超链接颜色改变，关键代码如下：

```
A:hover {
    Color:blue;
}
```

3.3.5 其他选择器

CSS 选择器除了以上几类外，还可以对选择器进行组合使用，限定风格作用范围。其代码如下：

```
<style type="text/css">
h2 span{
Text-decoration:underline;
}
li.newsid{
font-size:36px
}
li#newslst{
font-size:12px
}
</style>
.....
<h2 class="newsid">学院<span>新闻</span></p>
```

```

<ul>
<li class="newslid">秦皇岛校外学习中心</li>
<li id="newslid">秦皇岛校外学习中心</li>
<li>秦皇岛校外学习中心</li>
</ul>

```

说明:

(1) **h2 span:** 包含选择器 (也称上下文选择器), 用来选择特定元素或元素组的后代, 将对父元素的选择放在前面, 对子元素的选择放在后面, 中间加一个空格分开。后代选择器中的元素不仅仅只能有两个, 对于多层祖先后代关系, 可以有多个空格加以分开。

(2) **li.newslid, li#newslid:** 组合选择器, 标签选择器可以与类名选择器或 ID 选择器进行组合, 设置特定内容的 CSS 风格。

3.4 CSS 样式表类型

根据风格作用范围和效果的需求, CSS 样式表可分为: 行内 CSS、内嵌 CSS 和外部 CSS 三大类。通过不同 CSS 样式表的应用, 编程人员可以灵活、方便地针对网页中某一内容的具体风格, 某个网页甚至整个网站的若干个网页的统一风格进行设置。

3.4.1 行内样式

行内样式是直接 HTML 标签上定义该标签的 CSS 样式, 通过使用属性 `style` 来设置, 定义要显示的风格, 这是最简单的样式定义风格。不过, 通过此方法定义的风格, 只能控制此标签, 其语法如下:

```
<p style="color:#FF0000;font-family:宋体;"></p>
```

3.4.2 内嵌样式

内嵌样式是把样式表放在 `<style></style>` 代码块中, 并且置于 `<head></head>` 标签中, 定义的样式表可用于整个网页。其语法如下:

```

<head>
<style type="text/css">
    选择器 1{属性:属性值; 属性:属性值;...}
    选择器 2{属性:属性值; 属性:属性值;...}
    ...
    选择器 n{属性:属性值; 属性:属性值;...}
</style>
</head>

```

3.4.3 外部样式

将 CSS 规则定义放在单独的文件中 (.css), 然后在 HTML 页面中通过编写代码与 CSS 文件链接。

1. 代码链接方式

在网页中使用 `<link>` 标记链接到样式表, `<link>` 标记必须放到 `<head></head>` 区块中, 其格

式如下：

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

<link>标记表示浏览器从“外部样式表文件.css”文件中以文件格式读出定义的样式表。
rel="stylesheet"表示在网页中使用外部 CSS 文件；href="外部样式表文件.css"用于定义 CSS 文件的 URL；type="text/css"定义在 href 属性中指定的 MIME 文件类型为样式表文本。

2. 代码导入方式

使用@import 方式将外部 CSS 文件所定义的规则输入到 HTML 页面中，放置在<style></style> 标记中。其格式如下：

```
<style>
  @import url(外部样式表文件.css)
</style>
```

行内样式表、内嵌样式表、外部样式表各有优势，在实际的开发中常常需要混合使用。如果有整个网站统一风格的样式代码，应放置在独立的样式文件 (*.css) 中，某些样式不同的页面，除了链接外部样式文件，还需定义内嵌样式；某张网页内，部分内容“与众不同”，采用行内样式。

此外，对于某个 HTML 标签，如果有多种样式：

(1) 如果规定的样式没有冲突，则叠加。

(2) 如果有冲突，则最先考虑行内样式表显示，如果没有，再考虑内嵌样式显示，如果还没有，最后采用外面样式表显示，否则就按 HTML 的默认样式显示。