# 第3章

# CSS3基础入门



对于网页设计而言, CSS 就像一支画笔, 可以勾勒出优美的画面, 它可以根据设计者的要求对页面的 布局、颜色、字体、背景和其他图文效果进行控制, 可以说 CSS 是网页设计中不可缺少的重要内容。

# 🝚 重点导读

·了解 CSS 样式表。

- · 熟悉 CSS 样式表的基本语法。
- ·掌握 HTML 网页应用 CSS 样式的方法。
- · 熟悉 CSS 三大特性。
- · 熟悉在脚本中修改显示样式。

## 3.1 CSS 概述

CSS 样式的使用,减少了在 HTML 中重复的格式设置,如网页的颜色、字体和大小等。另外,在后期的维护中,如果需要修改部分外观样式,只需要修改相应的代码即可。

### 3.1.1 CSS 概述



CSS 被称为层叠样式单,它是一种专门描述结构文档的表现方式的文档,主要用于网页风格设计,包括字体大小、颜色、背景以及元素的精确定位等。在传统的 Web 网页设计里,使用 CSS 能让单调的 HTML 网页更富表现力。

CSS 样式的使用,使"网页结构代码"和"网页格式风格代码"分离开,将站点上的所有网页都指向 某个 CSS 文件,遇到要修改时,只需更改该 CSS 文件就可以了。不仅维护简单,还可以使 HTML 文档代 码简练,缩短浏览器的加载时间。W3C 组织也大力提倡使用样式单来描述结构文档的显示效果。 HTML 5+CSS 3+JavaScript 从入门到项目实践(超值版)

## 3.1.2 CSS 的发展历史

CSS 的最新版是 CSS 3,目前正处于进一步完善中。下面简单地介绍一下 CSS 的发展历史。

- CSS 1.0: 1996 年 12 月, CSS 1.0 作为第一个正式规范面试, 其中已经加入了字体、颜色等相关属性;
- CSS 2.0:1998 年 5 月, CSS 2.0 规范正式推出,这个版本的 CSS 也是最广为人知的一个版本,以前 的前端开发者使用的一般就是这个 CSS 规范;
- CSS 2.1:2004 年 2 月, CSS 2.1 对原来的 CSS 2.0 进行了一些小范围的修改,删除了一些浏览器支持不成熟的属性,我们可以认为 CSS 2.1 是 CSS 2.0 的修订版;
- CSS 3:2010年 CSS 3 规范推出,这个版本完善了前面 CSS 存在的一些不足,例如:颜色模块增加 了色彩校正、透明度等功能,还增加了变形和动画模块等。



## 3.1.3 CSS 3 的新功能

CSS 3 新增了许多的新功能,以前很多效果都需要借助脚本或者图片才能实现,使用 CSS 3 只需要几 行代码就能搞定了,具体的一些新功能如表 3-1 所示。

| 新功能  | 说 明  |
|------|--|
| 选择器  | CSS 3 增加了许多更强大的选择器                               |
| 边框   | CSS 3 可以创建圆角边框、阴影、边框背景等                          |
| 文字效果 | 使用 CSS 3, 设计者可以使用自己喜欢的任何字体,只需把自己喜欢的字体引入网站中就可以实现了 |
| 背景   | CSS 3 背景包含了新属性,包括背景图片的大小、裁剪背景图片、背景图片的定位等         |
| 渐变   | CSS 3 定义了线性渐变和径向渐变                               |
| 多列布局 | 为页面布局提供了更多的手段                                    |
| 动画   | CSS 3 动画使得设计者不需要编写脚本代码,也可以让页面元素动起来               |
| 媒体查询 | 可以根据不同的设备、不同的屏幕来调整页面                             |

表 3-1 CSS 3 新增功能



# 3.2 CSS 的基本语法

CSS 的语法非常简单, CSS 语法规则由两个主要的部分构成, 分别是选择器, 以及一条或多条声明, 具体语法结构如图 3-1 所示。



选择器通常是用户需要改变样式的 HTML 元素,选择器直接与 HTML 代码对应,声明(declaration) 非常人性化,属性(property)是用户希望设置的样式属性(style attribute),绝大部分属性名都是有含义的 英文单词或词组,每个属性对应一个值,属性值大部分也是直接用有意义的单词表示,例如颜色值可以取 blue、red 和 yellow,预设的 border 样式有 solid 和 dashed,属性和值之间用冒号分开。

CSS 语法具有很高的容错性,即一条错误的语句并不会影响之后语句的解析,代码如下:

```
<style>
h1{
    color:blue /*这里没有分号,导致语法错误*/
    font-size:20px /*这条声明不会被应用*/
}
h2{
    -color:red; /*对于不识别的属性名,CSS将自动忽略*/
    font-size:22px; /*前面的错误不影响这条声明的作用*/
}
</style>
```

注意:虽然 CSS 的容错性非常高,但是在编写的过程中也要注意语法错误的检查,用户可以使用 CSS Lint、Dreamweaver 等工具来检查 CSS 语法格式。

## 3.3 HTML 网页应用 CSS 样式的方法

CSS 可以控制 HTML 文档的显示,但在控制文档显示之前,需要在文档中引入 CSS 样式,HTML 提供了4种引入方式,包括行内样式、内嵌样式、链接样式和导入样式。

### 3.3.1 使用行内样式表



行内样式是最为简单的 CSS 设置方式,需要给每一个标签都设置 style 属性。它和样式所定义的内容在同一代码行内,通常用于精确控制一个 HTML 元素的表现,代码如下:

行内样式表

【例 3-1】(实例文件: ch03\Chap3.1.html)行内样式表。

相关的代码实例请参考 Chap3.1.html 文件,在 IE 浏览器中运行的结果如图 3-2 所示。

HTML 5+CSS 3+JavaScript 从入门到项目实践(超值版)



图 3-2 行内样式表作用效果

这种样式表不经常用, CSS 样式与 HTML 结构没有分离,导致代码冗余,并且不利于维护。



# .3.2 使用内部 CSS

内部 CSS 样式表一般是将 CSS 写在<head></head>标签中,并使用<style></style>标签进行声明,代码如下:

<head> <style> ... </style> </head>

【例 3-2】(实例文件: ch03\Chap3.2.html)内部样式表。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
      p{
         color: blue;
                             /*设置 p 标签的字体颜色*/
         font-size: 20px;
                             /*设置 p 标签的字体大小*/
         font-family: 宋体;
                             /*设置 p 标签的字体类型*/
      }
   </style>
</head>
<body>
  <使用内部 CSS 样式表</p>
  <使用内部 CSS 样式表</p>
  使用内部 CSS 样式表
</body>
</html>
```

相关的代码实例请参考 Chap3.2.html 文件,在 IE 浏览器中运行的结果如图 3-3 所示。



图 3-3 内部样式表作用效果

## 3.3.3 引入外部样式表



引入外部样式表是使用频率最高、也是最为实用的方法。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。

引入外部样式表是指在外部定义 CSS 样式表并形成以".css "为扩展名的文件,然后在页面中通过<link>引入页面中,代码如下:

```
<link rel="stylesheet" href="style1"/>
```

- rel:指定引入样式表,其值为 stylesheet。
- href 指定了 CSS 样式表的位置,此处表示当前路径下名称为 style1.css 文件。
- 【例 3-3】(实例文件: ch03\Chap3.3.html)引入外部样式表。

引入的外部样式表 style1.css 代码:

p{

```
color: green; /*设置 p 标签的字体颜色*/
font-size: 20px; /*设置 p 标签的字体大小*/
font-family: 隶书; /*设置 p 标签的字体类型*/
```

相关的代码实例请参考 Chap3.3.html 文件,在 IE 浏览器中运行的结果如图 3-4 所示。



图 3-4 引入外部样式表作用效果

## 3.3.4 导入外部样式文件



导入外部样式文件是指在内部样式表的<style>标记中,使用@import 导入一个外部样式表,代码如下:



```
<head>
<style>
@import "style2.css"
</style>
</head>
```

【例 3-4】(实例文件: ch03\Chap3.4.html)导入外部样式表。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
     /*导入外部的样式表 style2.css 文件*/
     @import "style2.css";
  </style>
</head>
<body>
导>导入外部样式文件
导>导入外部样式文件
<中>导入外部样式文件
</body>
</html>
```

在上面的代码中导入的外部样式文件 style2.css 代码:

```
p{
color: orange; /*设置p标签的字体颜色*/
font-size: 20px; /*设置p标签的字体大小*/
font-family:华文隶书; /*设置p标签的字体类型*/
```

相关的代码实例请参考 Chap3.4.html 文件,在 IE 浏览器中运行的结果如图 3-5 所示。



图 3-5 导入外部样式表作用效果



如果在开发 CSS 中遇到需要特别说明的地方,可以使用 CSS 注释进行注解说明,有利于其他程序员理 解你开发的 CSS 代码。

要注释 CSS 样式表,只需要在注释的内容前使用 " /\* "标记开始注释,在内容的结尾使用 " \*/ "结束 注释。代码如下:

```
<style>
p{
color: blue; /*设置 p标签的字体颜色*/
font-size: 20px; /*设置 p标签的字体大小*/
font-family: 宋体; /*设置 p标签的字体类型*/
}
</style>
```

## 3.4 CSS 三大特性

CSS 三大特性——层叠性、继承性和优先性。

### 3.4.1 CSS 层叠性



层叠性就是处理 CSS 冲突的能力。当同一个元素被两个选择器选中时,CSS 会根据选择器的权重决定 使用哪一个选择器。权重低的选择器效果会被权重高的选择器效果覆盖掉,权重相同时取后者。

权重可以理解为一个选择器对于这个元素的重要性。id 选择器权重为 100, 类选择器权重为 10, 标签 选择器的权重为 1, 如图 3-6 所示。

【例 3-5】(实例文件: ch03\Chap3.5.html) CSS 层叠性。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
     #box p{
        color: yellow; /*设置#box p 元素的字体颜色为黄色*/
     #box .box{
        color:red;
                      /*设置#box .box 元素的字体颜色为红色*/
     }
  </style>
</head>
<body>
<div id="box">
  我是什么颜色呢?
</div>
</body>
</html>
```

相关的代码实例请参考 Chap3.5.html 文件,在 IE 浏览器中运行的结果如图 3-7 所示。

HTML 5+CSS 3+JavaScript 从入门到项目实践(超值版)



图 3-7 层叠性

# \_\_\_\_\_3.4.2 CSS 继承性

继承性是子元素继承父元素样式的特性,在CSS中以text-、font-、line-开头的属性以及 color 属性都可以继承。

有一些比较特殊的元素,如<a>标签的颜色不能继承,必须对<a>标签本身进行设置,<h>标签的字体 大小不能继承,必须对<br/>ch>标签本身进行设置。

【例 3-6】(实例文件: ch03\Chap3.6.html) CSS 继承性。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title</title>
   <style>
      div{
         font-size: 20px;
                                            /*设置标签的字体大小*/
         text-shadow: 2px 2px 2px #FF0000;
                                            /*设置标签的字体阴影*/
         color: gold;
                                            /*设置标签的字体颜色*/
      }
   </style>
</head>
<body>
<div>
  >p标签的继承性
  <a href="">a 标签的继承性</a>
   <h1>h 标签的继承性</h1>
</div>
</body>
</html>
```

相关的代码实例请参考 Chap3.6.html 文件,在 IE 浏览器中运行的结果如图 3-8 所示。



图 3-8 继承性

可以看到,我们只给 div 设置了样式,3个子元素都继承了父元素的样式,除了<a>标签没有继承颜色, <h>标签没有继承字体大小。

### 3.4.3 CSS 优先性



当不同的规则作用到同一个 html 元素上时,如果定义的属性有冲突,那么应该用谁的值?CSS 有一套 优先性的定义,顺序如下:

!important>行内样式>ID 选择器>类选择器>标签选择器>通配符>继承>浏览器默认属性值。 【例 3-7】(实例文件:ch03\Chap3.7.html)CSS 优先性。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
     #box1{
                           /*设置#box1 元素的字体颜色*/
       color: gold;
     }
     .box1{
                            /*设置.box1 元素的字体颜色*/
        color: red;
     }
     #box2{
       color: gold;
                            /*设置#box2 元素的字体颜色*/
     }
     .box2{
       color: red!important;
                           /*设置.box2元素的字体颜色,!important优先性最高*/
     }
  </style>
</head>
<body>
id 选择器大于类选择器的优先性
!important 优先性最高
</body>
</html>
```

相关的代码实例请参考 Chap3.7.html 文件,在 IE 浏览器中运行的结果如图 3-9 所示。



图 3-9 优先性



# 3.5 在脚本中修改显示样式

在很多情况下,需要使用脚本来动态控制页面的显示效果,实现起来也很简单,只需要在脚本中获取 到该元素,然后修改它的 CSS 样式就可以了。

## 3.5.1 随机改变页面的背景色

随机改变页面的背景颜色很简单,这里我们使用"rgb(a,b,c)"属性来实现,只需要随机生成 a、b、c 的值就可以了,a、b、c 取值在 0~256,具体请看下面的实例。

【例 3-8】(实例文件: ch03\Chap3.8.html)随机改变背景色。

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <title>Title</title>
</head>
<body id="body">
<input type='button' value='刷新页面,随机切换背景颜色'/>
</body>
</html>
<script>
      var r = Math.floor(Math.random() * 256);
                                                //随机生成 256 以内的值,赋值给变量 r
                                                 //随机生成 256 以内的值,赋值给变量 g
      var g = Math.floor(Math.random() * 256);
      var b = Math.floor(Math.random() * 256);
                                                 //随机生成 256 以内的值,赋值给变量 b
      var color='rgb(${r},${g},${b})';
                                                 //使用 ES6 语法拼接 rgb 格式颜色
      var bd=document.getElementById("body")
                                                 //获取页面中的 body 元素
      bd.style.background=color;
                                                 //设置 body 元素的背景为 color
```

</script>

相关的代码实例请参考 Chap3.8.html 文件,在 Chrome 浏览器中运行的结果如图 3-10 所示。每当刷新 页面时,页面背景颜色会随机改变,如图 3-11 所示。



图 3-10 页面加载完成时的效果



#### 图 3-11 刷新页面后的效果



## 3.5.2 动态增加立体效果

在 CSS 中,通过简单的色差来实现简单的立体效果。如下面的案例,分别设置 div 四边边框的颜色, 产生了一个色差,呈现出立体的效果。同时还添加了一个按钮,来动态控制立体效果。当单击按钮时, div 呈现立体效果。 【例 3-9】(实例文件: ch03\Chap3.9.html) 动态增加立体效果。

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
   <title>Title</title>
   <style>
      .show{
         width:150px;
                                           /*设置宽度*/
         height: 50px;
                                           /*设置高度*/
         text-align:center;
                                           /*设置文本水平居中*/
                                           /*设置垂直居中*/
         line-height:50px;
         border-right:#222222 20px solid;
                                          /*设置右边框*/
         border-bottom:#222222 20px solid; /*设置底边框*/
                                          /*设置左边框*/
         border-left:#dddddd 20px solid;
         border-top:#dddddd 20px solid;
                                          /*设置上边框*/
         background-color:#cccccc;
                                           /*设置背景颜色*/
      }
   </style>
</head>
<body>
<input type='button' value='动态增加立体效果'onclick="change()"/><hr/>
<div id="box">立体效果</div>
</body>
</html>
<script>
  //定义 change 函数
   function change(){
      document.getElementById("box").className="show"; //给 box 添加 show 样式
   }
</script>
```

相关的代码实例请参考 Chap3.9.html 文件, 在 IE 浏览器中运行的结果如图 3-12 所示。当单击"动态增加立体效果"时,页面显示效果如图 3-13 所示。

|                                      |        | × |
|--------------------------------------|--------|---|
| 29(7) 希望(8) 意動(V) 在里来(A) 工具(7) 年初(9) | -      |   |
| 动态增加立体效果                             |        |   |
|                                      |        | - |
| 立体效果                                 |        |   |
|                                      |        |   |
|                                      | # 160% |   |

图 3-12 页面加载完成效果



#### 图 3-13 单击按钮后效果

## 3.6 实践案例——设计登录和注册界面



在平时浏览网页时,经常会遇到让我们输入信息的界面,登录后才能查看一些信息,如果没有在该网站注册过,还需要先注册信息。本案例也来实现一个简单的登录注册界面。



### 设计 HTML 结构,代码如下:

```
<div class="content">
<div class="main">
<h2>You are welcome!</h2>
<form>
<input type="text" placeholder="用户名"/>
<input type="password" placeholder="密码">
<button type="submit">登&nbsp;&nbsp;录</button>
<button type="submit">注&nbsp;&nbsp;册</button>
</form>
</div>
```

可以发现,我们设置了一个标题、两个文本框和两个按钮,在 IE 浏览器中运行的结果如图 3-14 所示。

| -<br>()))))))))))))))))))))))))))))))))))) | 口           | × |
|--|-------------|---|
| 文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T)             | 帮助(H)       |   |
| You are welcome !                          |             |   |
| 用户名 密码                                     |             |   |
| 登录 注册                                      |             |   |
|  |             |   |
|  | <b>100%</b> | • |

图 3-14 没设置 CSS 的页面效果

然后给每个元素添加 CSS 样式,全部代码如【例 3-10】所示。 【例 3-10】(实例文件:ch03\Chap3.10.html)登录注册界面设计。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>登录和注册页面设计</title>
  <style>
     .content{
        background-color: #1cff89;
                                  /*设置背景颜色*/
        position: absolute;
                                     /*设置绝对定位*/
        width: 100%;
                                     /*设置宽度*/
        height: 400px;
                                     /*设置高度*/
     }
      .main {
        text-align: center;
                                    /*设置文本居中*/
        padding: 50px 0px;
                                    /*设置内边距*/
        margin: 0 auto;
                                    /*设置外边距*/
      }
     h2 {
                                  /*设置字体类型*/
        font-family:"微软雅黑";
        font-size: 40px;
                                     /*设置字体大小*/
        font-weight: bold;
                                     /*设置字体加粗*/
     input {
```

```
/*设置边框*/
          border: 1px solid white;
         display: block;
                                           /*设置 input 为块级元素*/
         margin: 10px auto;
         padding: 5px;
         width: 230px;
         border-radius: 15px;
                                           /*设置圆角边框*/
         font-size: 18px;
                                           /*设置字体加粗*/
         font-weight: 300;
         text-align: center;
      button {
         background-color: #20a7ff;
         border-radius: 10px;
         border: 0;
         height: 30px;
         width: 80px;
         padding: 5px;
         color: white;
                                           /*设置字体颜色*/
      }
   </style>
</head>
<body>
<div class="content">
   <div class="main">
      <h2>You are welcome ! </h2>
      <form>
         <input type="text" placeholder="用户名"/>
         <input type="password" placeholder="密码">
         <button type="submit">登&nbsp;&nbsp; $/button>
             <button type="submit">注&nbsp;&nbsp;册</button>
         </form>
   </div>
</div>
</body>
</html>
```

### 相关的代码实例请参考 Chap3.10.html 文件,在 IE 浏览器中运行的结果如图 3-15 所示。



图 3-15 登录和注册完成界面



# 3.7 就业面试技巧与解析

## 3.7.1 面试技巧与解析(一)

面试官:在加载 CSS 文件时, link 引入外部样式和@import 导入外部样式有什么区别?

应聘者:link 与@import 在显示效果上还是有很大区别的,link 的加载会在页面显示之前全部加载完,而 @import 是读取完文件之后再加载,所以,在网络速度很好或很快的情况下,会出现刚开始没有 CSS 定义, 而后才加载 CSS 定义,@import 加载页面时开始的瞬间会有闪烁(无样式表的页面),然后恢复正常(加载 样式后的页面),link 没有这个问题。所以推荐使用 link 引入外部样式。

## 3.7.2 面试技巧与解析(二)

面试官:CSS hack 是什么?

应聘者:CSS hack 是根据不同的浏览器编写不同的 CSS 样式。由于不同厂商的浏览器,例如 IE、 Firefox、Chrome 等,或者是同一厂商的浏览器的不同版本,如 IE 6 和 IE 7,对 CSS 的解析认识不完全一 样,会导致生成的页面效果不一样,得不到我们所需要的页面效果。这个时候我们就需要针对不同的浏览 器去写不同的 CSS,让它能够同时兼容不同的浏览器,能在不同的浏览器中显示我们想要的页面效果。