# 第3章 chapter 3

# CSS

在 HTML 发展之初,人们使用标签的特殊属性或特定的标签来实现网页中背景颜色, 字体的颜色、大小、字号、粗体、斜体等样式的设定。后来 CSS 的出现改变了这一状况。

CSS(Cascading Style Sheets),即层叠样式表,简称样式表。它是一种用来表现 HTML或可扩展标记语言(eXtensible Markup Language,XML)等文件样式的计算机 语言。

传统 HTML 只有少量标签可以设定外观属性,属性数量也是有限的。而 CSS 可以 对 HTML 中的任意标签进行修饰,既可以单独对某个标签进行修饰,也可以对一组标签 进行修饰,非常灵活。它可以定义所选中标签的大量外观属性,使得网页的外观更加自 由、灵活。

CSS包含"层叠"之意,这意味着它可以对同一个标签进行多次设定,后面的设定会 覆盖前面的设定。这种设定在实际设计中非常实用。

# 3.1 CSS 语法

CSS有3种常用的引入方式:内联样式表、内部样式表和外部样式表。

#### 1. 内联样式表

内联样式表是通过为标签添加 style 属性的方式来实现引入的。例如:



上述代码中,标签添加了 style 属性,对它的字体颜色、背景颜色、字体大小和 字体样式进行了设置。其中,"color:red"意为将字体颜色设置为红色,"backgroundcolor:blue"意为将背景颜色设置为蓝色,"font-size:40px"意为将字体大小设置为 40 像 素,"font-family:楷体"将字体样式设置为楷体。其显示 效果如图 3-1 所示。

可以看到, 描述每个属性都是按照"属性名: 属性 图 3-1 内联样式表显示效果 值"的方式进行的。其他引入 CSS 描述属性的方式都是相同的。

#### 2. 内部样式表

内部样式表是通过添加<style>标签的方式来设置标签样式的。例如:

```
<!DOCTYPE html>
<html>
<head>
   <title></title>
   <style>
      p{
          border:1px solid red;
          width:400px;
          height: 50px;
          font-size: 40px;
          font-weight: bold;
          color:#336688;
       }
   </style>
</head>
<body>
   hello
   css
</body>
</html>
```

上述代码中,在<head>标签中加入了<style>标签。在<style>标签内,花括号 以外是选择器,决定这个样式对谁进行修饰,在这里是对所有标签进行修饰,这叫 作标签选择器。其代码格式如下:

选择器 { 属性名: 属性值; }

上例中,通过 border 属性将边框设置为 1px、实线、红色;通过 width 和 height 属性 将宽度和高度设置为 400px 和 50px;通过 font-size 属性将字体大小设置为 40px;通过 font-weight 属性将字体设置为粗体;通过 color 属性将字体颜色设置为 RGB 值 #336688,这种设置前两位 33 为红色分量,中间两位 66 为绿色分量,最后两位 88 为蓝色 分量。这就是 RGB 颜色表示法,即 # ff0000 红色、# 00ff00 为绿色、# 0000ff 为蓝色,其



他值为3种颜色的混合色。上述代码的显示效果如图 3-2 所示。

3. 外部样式表

hello					
css					
图 3-2	内部样式表显示效果				

外部样式表将 CSS 属性的描述存储在一个单独的 文件中,然后将这个文件引用到 HTML 中。先建立一 个 CSS 文件,命名为 outcss.css。代码如下:

```
div{
   width:400px;
   height:400px;
   float:left;
   background-color: #aa4466;
   color:white;
   margin: 10px 10px 10px;
   text-align: center;
   line-height: 400px;
   font-size: 80px;
```

}

可以看到在这个 CSS 文件中,样式的描述方式与内部样式表的描述方式并无不同。 这里设置了 div 的宽度和高度均为 400px;设置了 float 为 left,这使 div 可以从左向右排 满一行,如果不设 float 属性,每个 div 都会占一行,随后设置了 div 的背景颜色和字体颜 色;使用 margin 设置 4 个边距,依次为上边距、右边距、下边距、左边距;使用 text-align 设置字体为居中对齐;使用 line-height 设置行高为 400px,与 div 相等,这样字体会垂直 居中;使用 font-size 设置字体大小为 80px。最后建立一个 HTML 文件,引用这个 CSS 文件。

```
<! DOCTYPE html>
<html>
<head>
   <title></title>
   k rel="stylesheet" type="text/css" href="outcss.css">
</head>
<body>
   <div>Hello</div>
   <div>Hello</div>
   <div>Hello</div>
   <div>Hello</div>
   <div>Hello</div>
   <div>Hello</div>
   <div>Hello</div>
   <div>Hello</div>
</body>
```



</html>

上述代码中包含 8 个 < div > 标签, 使用 < link > 标签引用了 outcss.css 这个外部样 式表。图 3-3 为显示效果, 也可以用内部样式表的方式试一下, 其显示效果与外部样式表 没有任何不同。

这 8 个<div>标签依次排列,当一行占满后,会换到下一行。拖动浏览器可以看到, 随着浏览器窗口大小的改变,每行的<div>标签数量也会随之改变。改变浏览器窗口宽 度后的显示效果如图 3-4 所示。

Hello	Hello	Hello	Hello			
Hello	Hello	Hello	Hello			
图 3-3 外部样式表显示效果						



图 3-4 改变浏览器窗口宽度后的显示效果

这种布局方式相比传统固定大小的排版方式更能适配不同宽度的屏幕。

# 3.2 CSS 选择器

前面的例子中使用标签选择器对标签进行选择和修饰,即内部样式表演示中使用 p 选中了所有标签,外部样式表演示中使用 div 选中了所有<div>标签。

在实际的开发中经常需要对某个或某几个标签进行修饰,而不是对所有的同类标签 进行修饰,这时就应该使用其他类型的选择器。

常用的选择器类型有以下 6 种:标签选择器、id 选择器、类选择器、属性选择器,以及 将这些选择器综合运用的分组选择器和派生选择器。

#### 3.2.1 标签选择器

标签选择器直接使用标签名作为选择器。例如:

```
p{
    width:100px;
}
div{
```

color:red;

}

在 CSS 中可以依次写多个标签描述。上面例子对所有 p 的宽度都设置为 100px,所 有 div 的字体颜色都设置为红色。

#### 3.2.2 id 选择器

id 选择器是以标签的 id 为选择标准的选择器。在 HTML 中,每个标签都可以加一个 id 属性,id 之间不允许重复。当然,如果在 HTML 中确实存在了 id 重复的现象,浏览器也能容忍,而且有时也不会出问题。但需要强调的是,id 重复是不正确的,所以在设计 HTML 时要避免 id 重复。

```
<!DOCTYPE html>
<html>
<html>
<head>
        <title></title>
        <style type="text/css">
            #h{
                color:red;
        }
        </head>
<body>
        <div id='h'>hello</div>
        <div >world</div>
</body>
</html>
```

在这个例子中,有两个<div>标签,其中一个<div>标签的 id 为 h。在<style>标 签中使用 # h 的方式选择了这个 id 为 h 的<div>标签。将其 颜色设置为红色。通过图 3-5 可以看出,只有 id 为 h 的标签变 红了,另外一个<div>标签没有变化。这个 CSS 确实选中了 id 为 h 的标签。 图 3-5 id 选择器演示

#### 3.2.3 类选择器

看到"类"这个字也许大家会想到面向对象,然而 CSS 中的类和面向对象并无关系, 就是指一组标签。那么,怎样指定一组标签呢?答案是给标签加一个 class 属性。例如:

```
<!DOCTYPE html>
<html>
<head>
<title></title>
```



```
<style type="text/css">
       .rr{
           font-size: 30px;
           font-weight: bold;
           color:red;
       }
       .bb{
           border:1px solid blue;
           width:100px;
       }
   </style>
</head>
<body>
   <span class='rr'>hello</span>
   <span>css</span>
   <div class='rr bb'>haha</div>
   <div class='bb'>hehe</div>
</body>
</html>
```

在这个例子中包含两个<span>标签和两个<div>标签。其中第一个<span>标 签和第一个<div>标签都包含类名 rr。两个<div>标签都包含 bb 类。第一个<div> 标签属于两个类 rr 和 bb。

这个<style>标签使用了比较正规的写法,包含了 type 属性,值为 text/css。其中, 使用.rr的方式选中 rr 这个类,使用.bb的方式选中 bb 这个类。 hello css 对 rr 类设置了字体大小,使用 font-weight 设置了粗体,同时设 haha 置了字体颜色;对 bb 类设置了边框和宽度。其显示效果如

hehe 图 3-6 所示。 通过演示可以看出,标记为 rr 类的 < span > 标签和 图 3-6 类选择器显示效果 <div>标签都具备 30px 的字体大小和红色粗体的属性,标记

为 bb 类的两个<div>标签都具备边框和宽度的属性。

综上,CSS类是可以跨越标签类型的,不同的标签类型可以设置同样的类。另外,一 个标签是可以从属于多个类的。

#### 3.2.4 属性选择器

属性选择器使用属性是否存在或属性的值来对标签进行选择。例如:

```
<!DOCTYPE html>
<html>
<head>
   <title></title>
   <style type="text/css">
```

30



```
[title]{
    font-size:40px;
    text-decoration: line-through;
    }
    [title=haha]{
    text-transform: uppercase;
    }
    </style>
</head>
<body>
<h2>Demo</h2>
<h1 title="hello world">hello world</h1>
<a title="haha" href="#">zzzz</a>
</body>
</html>
```

上述代码中,使用[title]选择了所有带有 title 的标签,这里为<h1>标签和<a>标签。使用[title=haha]选择了<a>标签。这里出现了两个新的 CSS 属性,textdecoration为文本修饰,可以添加上画线、删除线和下画 线。可以用 text-transform 来对标签内容进行一些转换, 如转换成大写、转换成小写或首字母大写。这里为转成大 写。其显示效果如图 3-7 所示。



## 3.2.5 分组选择器

分组选择器可以同时对多个选择项设置同样的属性。例如:

```
<!DOCTYPE html>
<html>
<head>
   <title></title>
   <style type="text/css">
      #np,div{
         font-style: italic;
         font-size: 40px;
      }
   </style>
</head>
<body>
   hello
   css
   <div>this is div</div>
</body>
```



</html>

这里使用"♯np,div"的方式选择了两组标签。第一 组使用了id选择器,选择了id为"np"的标签;第二组使用 了标签选择器,选择了所有的div。该CSS使用font-style 设置字体为斜体,使用font-size设置字体大小为40px在 两组中都生效了。其显示效果如图 3-8 所示。



图 3-8 分组选择器显示效果

3.2.6 派生选择器

派生选择器实现了"某个标签内的某标签"的选择。例如:

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
    <title></title>
    <title></title>
    <title></title>
    <title></title>
    <title></title>
    <title></title>
    <title></title>
    <title></title>
    <ti><style type="text/css">
        p span{
            font-size: 40px;
        }
        </style>
</head>
<body>
        hello <span>css</span>
        <span>java</span>
</body>
</html>
```

上述代码中有两个<span>标签,现在要选择标签中的<span>标签,所以用选择器"p span"来实现这一选择。还可以使用类似" # hello a"或".hello p"的方式配合 id 选择器和类选择器来实现该功能。

在上面的例子中我们已经接触到很多 CSS 属性。在实际学习中,通常需要一个 CSS 文档,方便我们在不知道或记不住的情况下查询。这部分内容过于繁杂,这里不一一列举。

# 3.3 盒子模型

HTML 中的标签有 3 种主要的显示模式: none 模式、block 模式和 inline 模式。其中, none 为不显示,即隐藏; block 具备宽度和高度可以用 CSS 进行设置; inline 的宽度和高度由内容决定,不能自己设定。图 3-9 包含 3 个重要的概念:外边距(margin)、边框(border)和内边距(padding)。盒子模型如图 3-9 所示。

外边距为边框外距离父标签和兄弟标签的距离。边框也有自己的宽度,可以将4个 方向的边框分别设置为不同的类型。内边距(padding)为边框到当前标签内容之间的





距离。

我们可以这样写:div的外边距的上边距(margin-top)、右边距(margin-right)、下边距(margin-bottom)和左边距(margin-left)分别为 0px、10px、20px 和 10px。内边距也是按照上、右、下、左的顺序编写的。我们也可以只写"margin:0px",表示所有外边距均为 0px。

```
div{
    margin:0px10px20px10px;
    padding:0px10px10px20px;
}
h1{
    display:block;
    margin:0px;
    padding:0px;
}
```

# 3.4 定位方式

一般的标签是按照它和它所在的父标签的位置关系及 margin、border 和 padding 来 定位的。但也存在其他定位方式,如使用 CSS 的 position 属性可以进行设定。

position 的取值有以下 5 种:第一种为 fixed,即相对于浏览器来定位(绝对定位),使用 left、right、top 和 bottom 来对它进行定位。如果设定了 left,那么它相对于浏览器的 左侧的位置就确定了;如果设置了 right,那么它相对于浏览器右侧的距离就确定了;top 和 bottom 的定位方法与 left 和 right 类似。第二种为 static,即默认值,就是按照它和父 标签的关系来定位。第三种为 sticky,即黏性定位,该定位方法基于用户滚动的位置。它 的行为 就像 "position: relative;", 而 当 页 面 滚 动 超 出 目标 区 域 时, 它 的 表 现 就像



"position:fixed;",会固定在目标位置。第四种 absolute,即生成绝对定位的元素,相对于 static 定位以外的第一个父元素进行定位。第五种 relative,即相对定位,相对于正常位 置进行偏移。

```
<!DOCTYPE html>
<html>
<head>
   <title></title>
   <style type="text/css">
       #big{
           width:100px;
           height: 1500px;
          background-color: red;
       }
       #fix{
           position: fixed;
           right: 50px;
          bottom: Opx;
          background-color: blue;
          width: 100px;
          height: 100px;
       }
   </style>
</head>
<body>
   <div id='big'></div>
   <div id='fix'></div>
</body>
</html>
```

上述代码中有一个高度为 1500px 的大 div,这使得页面产生了滚动。但滚动页面 时会发现,不论如何滚动,右侧的蓝色方块永远不动,这是因为右侧这个 div 是 fixed 定 位方式,是相对于窗口固定的,所以不会受到页面滚动的影响。绝对定位演示如图 3-10 所示。



图 3-10 绝对定位演示

### 3.5 导航条示例

下面使用一个示例对所学知识进行综合运用。导航条在各个网站上都是不可或缺的,下面就做一个导航条。首先需要建立标签结构。

```
<ulid='nav'>
<a href='http://www.baidu.com'>百度</a>
```

可以看到,这是一个无序列表,列表中包含列表项,而列表项中包含<a>标签。这 是非常典型的一个导航条。

然后为其编写 CSS,通常我们在做的时候会边写边看效果。

```
<style type="text/css">
    * {
       margin: 0px;
       padding: 0px;
    }
    #nav, #nav li{
       display: block;
       list-style: none;
    }
    #nav{
       height: 40px;
       background-color: #006688;
       color: white;
    }
    #nav li{
       width:100px;
       float: left;
    }
    #nav a{
       display: block;
       width: 100px;
       text-align: center;
       color: white;
```



```
line-height: 40px;
text-decoration: none;
}
#nav a:hover{
background-color: #661122;
}
```

</style>

首先使用 \* 选择所有标签,将其内外边距都设置为 0,这样便于消除导航条和顶端之间的缝隙。将标签和标签设置为 block 的显示方式,这样才能设置它们的宽度和高度。

将列表项的宽度设置为 100px,并设置为左浮动,这样就能排成一排。设置 nav 标签的高度为 40px,设置 nav 标签的背景颜色为灰色,设置 <a>标签为居中对齐,通过设置 行高实现垂直居中,设置 text-decoration 去掉下画线。

比较新鲜的一个写法是":hover",这是因为超链接有不同的状态,如未访问过、已访问过、鼠标放上、单击,CSS可以对这些状态设置不同的外观,实现一种动态效果。其中, "a:visited"为已访问过的外观,"a:hover"为鼠标放上时的外观,"a:active"为单击时的 外观。

导航条演示如图 3-11 所示,当鼠标放上时,列表项会变色。

百度	百度	百度	百度	百度	百度	
	J.					
www.baidu.com						

图 3-11 导航条演示

3.6 小 结

本章对 CSS 进行了简要的介绍。学习本章内容后,有助于读者开发一个有一定外观 效果的网站。如果想成为一个安全工程师,学习这些内容或许就够了。但是,如果想成 为一个合格的前端工程师,还有大量的内容需要自行学习。